
firefighter Documentation

Release 0.1.0

Kang Hyojun

Apr 18, 2019

Contents

1	Firefighter	1
1.1	Modules	1
1.2	Indices and tables	2
	Python Module Index	3

AWS Kinesis Data Firehose logging handler.

1.1 Modules

1.1.1 firefighter — Firehose logging handler

`firefighter.logging` — Firehose logging handler

```
import logging

from firefighter.logging import FirehoseHandler

logger = logging.getLogger('foo')
handler = FirehoseHandler(delivery_stream_name='bar')
logger.addHandler(handler)
logger.warning({'message': 'Ahoy'})
```

```
class firefighter.logging.FirehoseHandler(delivery_stream_name=None,
                                          use_queues=True,          send_interval=60,
                                          boto3_session=None,
                                          boto3_profile_name=None, *args, **kwargs)
```

Put logs on [AWS Kinesis Data Firehose](#).

Parameters

- **delivery_stream_name** – A firehose delivery stream name.

- **use_queues** – Deliver a log data with thread.
- **send_interval** – At least seconds to send a log data.
- **boto3_session** – An instance of `boto3.session.Session`.
- **boto3_profile_name** – A name of boto3 profile name.

MAX_BATCH_SIZE = 5120

A max batch size in bytes.

MAX_RECORD_SIZE = 1024

A max record size in bytes.

batch_sender (*batch_queue*, *delivery_stream_name*, *send_interval*)

Submit queued messages to Firehose. If one of following conditions is corresponded.

- `FirehoseHandler.close()` is called. so `FirehoseHandler.END` is queued in `batch_queue`.
- `FirehoseHandler.flush()` is called. so `FirehoseHandler.FLUSH` is queued in `batch_queue`.
- The data size of `batch_queue` is greather then `FirehoseHandler.MAX_BATCH_SIZE`.
- The count of `batch_queue` is greather then `FirehoseHandler.MAX_BATCH_COUNT`.
- The time of processing of `batch_queue` takes more seconds than `send_interval`.

Parameters

- **batch_queue** – A queue which is stored a log data to send it.
- **delivery_stream_name** – A firehose delivery stream name.
- **send_interval** – At least seconds to send a log data.

close()

Close handler.

emit (*message*, *make_thread*=<class 'threading.Thread'>)

Emit a log message. Queuing a message unless `FirehoseHandler.use_queues` is *false*. So that making an HTTP request is executed on another thread.

Parameters

- **message** – A message to be logged.
- **make_thread** – A callable object to create thread.

flush()

Flush messages.

exception `firefighter.logging.FirehoseWarning`

Firehose warning class. Every warning raised on `FirehoseHandler` should use this class.

1.2 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

f

`firefighter`, [1](#)

`firefighter.logging`, [1](#)

B

`batch_sender()` (*firefighter.logging.FirehoseHandler* method), 2

C

`close()` (*firefighter.logging.FirehoseHandler* method), 2

E

`emit()` (*firefighter.logging.FirehoseHandler* method), 2

F

`firefighter` (module), 1

`firefighter.logging` (module), 1

`FirehoseHandler` (class in *firefighter.logging*), 1

`FirehoseWarning`, 2

`flush()` (*firefighter.logging.FirehoseHandler* method), 2

M

`MAX_BATCH_SIZE` (*firefighter.logging.FirehoseHandler* attribute), 2

`MAX_RECORD_SIZE` (*firefighter.logging.FirehoseHandler* attribute), 2